

# Tailoring DoDAF For Service Oriented Architectures: A Recommended Guide

*Fatma Dandashi*

*Huei-Wan Ang*

The MITRE Corp.

7515 Colshire Dr.

McLean, VA 22102

703-983-7914, 703-983-6701

dandashi at MITRE.org, hwang at mitre.org

Keywords:

Service, Architecture, Service Provider, Service Consumer, Service Specification, Service, Department of Defense Architecture Framework

**Abstract:** *This paper provides guidance on how to use the Department of Defense Architecture Framework (DoDAF) to describe a Service-Oriented Architecture (SOA). With acceptance of Network-Centric Warfare (NCW) and “net-centricity”, the Department of Defense (DoD) is moving towards a dynamic producer/consumer environment. SOA is an architectural design approach to application integration that provides such an environment through the flexible connectivity of applications implemented as services. Such services have well-defined, platform-independent specifications that hide the underlying technical complexity of the implementation (encapsulation), are self-contained (loosely coupled), and reusable. DoDAF does not prescribe any particular approach (SOA or otherwise) for describing the architecture, only the elements and relationships needed to describe an architecture. Tailoring of the DoDAF elements and relationships, which DoDAF allows, is needed to meet the SOA paradigm shift. Further, creating DoDAF architecture descriptions that are service oriented supports globalization and the integration of geographically dispersed organizations (Net-centricity).*

## 1 Introduction <sup>1</sup>

The principal objective of the DoD Architecture Framework (DoDAF) [1] is to ensure that architecture descriptions can be compared and related across organizational boundaries, by defining a particular set of architectural elements and relationships used for describing architectures. Several tailorings/adaptations of DoDAF for use during the capability-based analysis process [2], for capturing the conceptual detail of various simulation schemes to reduce the burden of transition from specification to executable system ready for experimentation [3], and for describing the architecture of modeling and simulation systems to facilitate the certification, validation, and accreditation [4] have been defined. With acceptance of Network-Centric Warfare

[5] and “net-centricity”, the Department of Defense (DoD) is moving towards a dynamic producer/consumer environment. A Service Oriented Architecture (SOA) is an architectural design approach where application design and development is based on the concept of services. Because DoDAF does not prescribe any particular architectural approach, one can utilize DoDAF to describe a SOA. However, some tailoring is required to better support SOA design patterns within DoDAF. In this paper, we tailor DoDAF for defining reusable and composable services<sup>2</sup> and describing the resulting SOA. Our premise is that using DoDAF to describe a SOA enables leveraging the existing body of knowledge and architecture artifacts within DoD. Further, tailoring DoDAF for SOA enables architects to more effectively describe a SOA as an alignment of services to operational activities. The resulting architecture description is an

---

<sup>1</sup> This work was partially supported by the US Space Situation Awareness Integration Office and OSD’s Open Systems Joint Task Force – <http://www.acq.osd.mil/osjtf/#> – to be published at SIW conf: <http://www.sisostds.org>

---

<sup>2</sup> While the scope of this paper is software services, the DoDAF tailored products and underlying metamodel are equally applicable to human functions (e.g., as in outsourcing business services)

integrated model of a set of net-centric collaborating humans and systems services.

## 2 Definitions and Concepts

### 2.1 What is a Service?

The term *service* has been defined by several industry and standards consortia such as IBM [6], OASIS [7], OMG [8], etc., with slightly different variations. In general, the current body of knowledge is consistent in applying the term to a certain kind of **software** application with the following characteristics. A service provides well defined, self-contained functionality that is loosely coupled from other functionality/services. The *service* functionality is well encapsulated. That is, complexity of the implementation is hidden from potential consumers except for the information required by the consumers to determine whether a given service is appropriate for their needs; that information is exposed in the *service interface (specification)* [7], accessible through a *service port*. The semantics of a service should be documented, either directly or indirectly, by its specification or set of messages [9]). Services have coarse granularity, they tend to use a small number of operations with relatively large and complex messages which are exchanged between the *provider and consumers*. A service is location transparent (i.e., consumers do not need to be aware of the physical location of a hosting server); and protocol independent (messages are sent in a platform-neutral, standardized format delivered through the service specification). Services tend to be oriented toward use over a network, though this is not an absolute requirement. [9]

### 2.2 What is SOA?

SOA is a form of distributed systems architecture based on services (as defined above) where a consumer does not need to know the internal structure of a provided service, including features such as its implementation language, process structure, and even database structure [9]. In SOA, the focus is on the sequence of operational activities or business process. The process is then mapped to a systems architecture description with specific applications that support the process cast as services. Thus, the architecture supports an operational/business process via a set of independent, reusable, but collaborative services. The service integration happens dynamically, via service composition (the execution of several of these independent services in an orchestrated manner) [10].

### 2.3 Advantages of SOA

SOA offers several advantages:

1. An operational process orchestrates simple services into complex services
2. Services allow the exchange of information and data between:
  - a. different computers, from different vendors,
  - b. different programs, from different functional areas, or different members of a Community of Interest (CoI),
3. SOA supports globalization and the integration of geographically dispersed organizations (net centricity) through service orchestration of distributed services owned and executed across ownership boundaries [6].

## 3 Tailoring DoDAF for SOA

### 3.1 Rational for Tailoring DoDAF for SOA

The argument for tailoring DoDAF for SOA is that it enables leveraging the existing body of knowledge and architecture products within DoD. Tailoring DoDAF for SOA helps ensure consistency among operational activities specified in the Operational View (OV) and the services specified in the Systems View (SV). This tailoring enables architects to more effectively describe SOA as an alignment of services to operational activities and to identify common functionality as a set of re-usable services.

Arguments for not using DoDAF are based on several misconceptions. For example, there is a misconception that DoDAF enforces description of a “Point-to-Point” architecture. DoDAF is intended to show required key information flows and interfaces (e.g., Key Interface Parameters-KIPs) in the OVs which map to well-defined data flows between services. Further, neither DoDAF nor SOA require a description of all possible information flows or interfaces.

### 3.2 Tailoring Details

The vocabulary used in this paper is based on an internal MITRE effort<sup>3</sup> to evolve DoDAF and to define a clear delineation between the requirements submodel: the **Operational or Resource View**, and a solution submodel: the **Systems View** which is further divided into two subviews: the **Automated System** and the **Human** subviews. An *Operational Resource* is defined as an actor whose responsibilities are allocated in the SV to

---

<sup>3</sup> With contributions from David Nicholson of MITRE

*Humans and Automated Systems* (i.e., hardware and software). As part of this MITRE effort, a metamodel has been defined to support tailoring DoDAF for SOA. The tailoring involves introducing some new elements and relationships, as well as modifying the composition of elements and relationships for some products.

### 3.2.1 Elements

The new elements are *service port*, *service specification*, and *service requirement* or contract. Definitions for these new elements are provided below.

**Port.** A distinct interaction point between an element of the architecture and its environment. It specifies the services provided as well as the services required by that element. Service specifications are provided and required through ports in order to decouple consumers and providers [7, 8].

**Service Specification.** A description of what service consumers need to do in order to use a service, and what service providers do in order to implement the service [7, 8].

**Service Requirement (Contract).** A statement(s) of functionality that a service must meet. It includes a description of the participants in the contract and the *roles* they play. A *Service Level Agreement (SLA)* is a formal representation of the service requirements or contract and specifies performance requirements [7, 8].

Other elements needed for SOA are *service*, *service data* or real world effect, and *service standard* [7, 8]. These elements are not new to DoDAF. They are specializations of a system function, systems data, and technical standard, respectively.

### 3.2.2 Relationships

Relationships between the new elements and existing DoDAF elements have been formally defined in an integrated metamodel. The diagrams in Appendix B (see Figure 7 and Figure 8) represent excerpts from the metamodel. The excerpts show the elements and relationships relevant for SOA. Elements that are new to DoDAF v 1.0 are highlighted in Blue. Relationships to and from these new elements constitute new relationships.

### 3.2.3 Products

No tailoring of the OV is required to support SOA. However, one of the tenets of DoDAF is to rigorously define the OV products; and one of the promises of SOA is that it allows for better alignment of applications with operational processes. In order to accomplish this

alignment, SOA requires that the operational processes be both well defined, and defined at a granular enough level to be able to map directly to the services in the Systems View. The OVs are a conceptual model consisting of OV-2s, OV-5s, and OV-6s. An operational node is a logical grouping or specification of a certain kind of community of interest that provides or requires a service(s). At this level, a generic specification of required or provided operational activities (that may be performed by humans and/or machines) is described, and assigned to these logical nodes (CoIs). This does not impact current DoDAF definitions of OVs.

SV products need to be tailored to include the new elements and relationships defined above as well as to define new compositions. A logical SV model, using a set of SV-1, SV-4, and SV-10 products is used to define services, and their interactions through service specifications, with OV-7 and possibly SV-11s, to define the data structures for the domain. In addition, report products such as SV-3, SV-5, SV-6, and SV-7 are tailored in a new format to show a matrix of relationships among elements that are defined in SV-1 and SV-4.

Specifically, a systems node in a logical SV-1 is a specification of a logical deployment node (not a physical instance, and not tied to a geographic location), such as a logical specification of a generic service hosting center, or group of service consumers. This does not impact current DoDAF definitions for SV-1 but clarifies the current definition of SV-1 Systems nodes. At this logical level, there is no need to model systems. Services and service consumers may be grouped by systems nodes as shown in Figure 1. Figure 2 shows further detail, where systems and services are shown on systems nodes.

An SV-2 is used to specify physical nodes and the underlying network providing the infrastructure that enables communications between providers and consumers (allocation of SV-1 logical nodes to multiple physical instances of deployment nodes). An example SV-2 is not provided in this paper.

## 4 DoDAF with SOA Tailoring Example

### 4.1 Space Weather Impact Analysis Example Scenario

In this paper, tailored DoDAF products supporting the following scenario are provided as an example. Space weather information from the Space Weather Service is required to provide Space Situation Awareness (SSA) for Space Command and Control (C2). Space weather information is also required to support theater operation

planning and commercial satellite launch planning. A Space C2 Operator uses the Space C2 Application to view a User Defined Operational Picture (UDOP). In providing the UDOP display, the Space C2 Application uses the provided UDOP criteria to request the Space UDOP from the Space UDOP Service, which in turn sends a request to the Space Weather Service to provide a space weather impact report for a specified period of time. The Space Weather Service queries the Weather Data Base (DB). The WeatherDB returns the queried space weather impact report present at that moment in the database. If the space weather information requested is not in the database, the Space Weather Service sends a request to the Space Weather Impact Analyzer to make and provide a space weather impact prediction. The Space Weather Service returns a space weather impact report as the response to the Space UDOP Service, which integrates the report with all other SSA information and returns the requested Space UDOP to the Space C2 Application. The Space C2 Application displays the Space UDOP. The example provided does not represent a complete architecture description and each example product does not necessarily include the entire potential content of a product, but includes enough detail to properly demonstrate the tailoring of a product. In addition, the content of the example is based on information provided by the Space Situation Awareness Integration Office (SSAIO) but does not represent actual or planned operational processes or systems from Space Command.

#### 4.2 Relationship to UML

DoDAF v1.0 included guidance for representing DoDAF architecture products using UML. However, this guidance was not prescriptive and was based on UML 1.x, which is what was available at the time of writing DoDAF v1.0. Subsequently, different vendors created different implementations which lead to interoperability issues between UML modeling tools and imposed additional training requirements for users. The absence of an industry standard makes it difficult to build, reuse, merge, exchange, or compare architecture models in a collaborative manner. The growing interest in force planning for coalition operations, and the ability to share and relate architecture information developed by different nations using an industry architecture framework standard is becoming increasingly important.

With guidance from OSD, the Object Management Group (OMG) has issued a request to define a UML profile for DoDAF and UK's Ministry of Defense Architecture Framework (MoDAF) [11]. A consortium of industry leaders and UML tool vendors is currently working to

define this profile, which will help improve the ability to share and relate architecture information developed using different UML modeling tools. As a precursor to this profile, the example architecture provided here was developed using UML 2.0<sup>4</sup>. However, a SOA Request for Proposal [8] is still under development by OMG, and the UML 2 notation used here is not the OMG adopted specification for describing SOA using UML. See Appendix A for an overview of DoDAF with example products tailored for SOA.

## 5 Summary

SOA is an architectural approach where application design and development is based on the concept of services. The principal objective of DoDAF is to ensure that architecture descriptions can be compared and related across organizational boundaries, by defining a particular set of architectural elements and relationships used for describing architectures. Because DoDAF does not prescribe any particular approach, one can utilize DoDAF to describe a SOA. This paper defined how to tailor DoDAF to describe SOA by identifying specific SOA characteristics that must be added to the set of elements that DoDAF v1.0 defined, such as a service specification. Tailoring DoDAF for SOA enables architects to more effectively describe a SOA as an alignment of services to operational activities thus describing an architecture as a set of net-centric collaborating humans and systems services.

## References

- [1] Department of Defense Architecture Framework v1.0  
[http://www.defenselink.mil/nii/doc/DoDAF\\_v1\\_Des\\_kbook.pdf](http://www.defenselink.mil/nii/doc/DoDAF_v1_Des_kbook.pdf)  
[http://www.defenselink.mil/nii/doc/DoDAF\\_v1\\_Volume\\_II.pdf](http://www.defenselink.mil/nii/doc/DoDAF_v1_Volume_II.pdf)  
[http://www.defenselink.mil/nii/doc/DoDAF\\_v1\\_Volume\\_I.pdf](http://www.defenselink.mil/nii/doc/DoDAF_v1_Volume_I.pdf)
- [2] Atkinson, K., "Modeling and Simulation Foundation for Capabilities Based Planning." Proceedings IEEE Spring Simulation Interoperability Workshop, IEEE CS Press, 2004
- [3] Muguira, J., & Tolk, A., "Applying A Methodology To Identify Structural Variances In Interoperations," Journal of Defense Modeling and Simulation:

---

<sup>4</sup> The example was developed using Sparx Systems' Enterprise Architect. <http://www.sparxsystems.com/products/ea.html>

- Applications, Methodology, Technology (JDMS), Spring, 2006
- [4] Tolk, A., and Solick, S., "Using the C4ISR Architecture Framework as a Tool to Facilitate V&V for Simulation Systems within the Military Application Domain," Proceedings IEEE Spring Simulation Interoperability Workshop, IEEE CS Press, 2003
- [5] "Memorandum: Global Information Grid Enterprise Services (GIG ES)- Net-Centric Environment," OASD(NII), July 2004  
<http://www.defenselink.mil/nii/doc/docArchive.html>
- [6] M. Endrei, J Ang, A Arsanjani, S. Chua, P. Comte, P. Krogh, M. Luo, T. Newling, "Patterns: Service-Oriented Architectures and Web Services," IBM RedBooks, IBM 2004  
<http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246303.html?Open>
- [7] Organization for the Advancement of Structured Information Standards (OASIS), "Service Oriented Architecture (SOA) Reference Model," Public Review Draft 1.0, 10 February 2006,  
<http://www.oasis-open.org/committees/download.php/16587/wd-soa-cd1ED.pdf>
- [8] Open Management Group, SOA SIG,  
<http://SOA.omg.org>
- [9] World Wide Web Consortium, <http://www.w3.org/>
- [10] <http://www.zapthink.com/>
- [11] <http://syseng.omg.org/UPDM.htm>  
<http://www.omg.org/cgi-bin/doc?dtc/2005-09-12>
- <http://www.omg.org/cgi-bin/doc?c4i/06-05-04>  
<http://www.omg.org/cgi-bin/doc?c4i/06-05-03>  
<http://www.omg.org/cgi-bin/doc?c4i/2006-05-02>
- [12] "Service Specification Template (S300)," Global Information Grid Net-Centric Implementation Document, 20 July 2006 (V3.0)  
<https://gesportal.dod.mil/>

## Author Biographies

**FATMA DANDASHI** is currently leading an Object Management Group (OMG) effort to define a UML profile for the DoD Architecture Framework. Prior to this activity she supported the development of Air Force Enterprise Architecture for SAF/XC. Dr. Dandashi was task lead on the MITRE development team responsible for DoD Architecture Framework Version 1.0 (Volumes I and II), and currently serves on the working group developing DoDAF Version 2.0. Dr. Dandashi holds a Ph.D. in Information Technology from George Mason University, a Master of Science Degree in Computer Science from the University of Louisiana (Lafayette), and a Bachelor of Arts degree in Computers/Business Administration from the Lebanese American University.

**Huei-Wan Ang** is a Senior Software Systems Engineer with the MITRE Corporation. She is the primary contributor of a MITRE effort to define an Architecture Specification Model for the next evolution of a DoD Architecture Framework. She supported the development of DoD Architecture Framework Version 1.0. She has over 10 years experience in system and software architecting and object-oriented software analysis and development. She has an M.S. degree in Information Systems from The American University and a B.S. degree in Computer Science from George Mason University.

## Appendix A

### 6 DODAF Products Tailoring Example

The vocabulary used in this paper is based on an internal MITRE effort to evolve DoDAF and to define a clear delineation between the requirements submodel: the **Operational or Resource View**, and a solution submodel: the **Systems View** which is further divided into two subviews: the **Automated System** and the **Human** subviews. An *Operational Resource* is defined as an actor whose responsibilities are allocated in the SV to *Humans* and *Automated Systems* (i.e., hardware and software). The following subsections are organized by the DoDAF set of 26 products for easy reference. However, that does not imply that a SOA description based on this DoDAF tailoring is product centric. Rather, the tailoring is based on one integrated semantic model (see Appendix B).

## **6.1 All View (AV)**

There are some overarching aspects of an architecture description that relate to all three views. These overarching aspects are captured in the AV products. Two products are defined in the All-View, Overview and Summary Information (AV-1) and Integrated Dictionary (AV-2). These two products do not need to be tailored for SOA.

## **6.2 Operational View (OV)**

OV describes the capabilities, operational activities, and information exchanges required to conduct operations. A pure OV is materiel independent and is used to describe capability requirements. No tailoring of the OV is required to support SOA. However, the OV is important to SOA because it describes the required capabilities and justification for application investment, and identifies specific operational activities that can be automated via services in a SOA. One of the promises of SOA is that it allows for better alignment of applications with operational processes. In order to accomplish this alignment, SOA requires that the operational processes be both well defined, and defined at a granular enough level to be able to map directly to the services in the Systems View. One of the tenets of DoDAF is to rigorously define the OV products to better enable this alignment.

### **6.2.1 High-Level Operational Concept Graphic (OV-1)**

For a SOA, OV-1 sets the known mission context for the architecture being described in the remaining products by describing the mission and highlighting known consumers and service providers, and any interesting or unique aspects of operations. It presents a description of the interactions between the subject architecture and its environment. An example OV-1 is not provided in this paper.

### **6.2.2 Operational Node Connectivity Description (OV-2)**

The purpose of the OV-2 is to depict all logical nodes that are groupings of operational activities and resources (a generic term for human and/or system) responsible for the operational activities. OV-2 also shows required needlines between these nodes. An example OV-2 is not provided in this paper.

### **6.2.3 Operational Product Exchange Matrix (OV-3)**

OV-3 documents *required* information exchanges between operational activities. It is important to note here that identifying required exchanges is of great value, in that the exchange is as much a part of the operational process as the functionality, since it documents required dependencies among interacting operational activities. This product should be generated from the operational activity interactions documented in OV-5 and OV-6c. An example OV-3 is not provided in this paper.

### **6.2.4 Organizational Relationships Chart (OV-4)**

The OV-4 illustrates the relationships among organizations or resources in an architecture description. These relationships are relevant to show in an OV because they illustrate fundamental *roles* and their relationships. The OV-4 describes the organizational relationships of those providing functionality as well as those for whom the functionality is intended - i.e., end-users, CoIs, or user/governing groups. An example OV-4 is not provided in this paper.

### **6.2.5 Operational Activity Model (OV-5)**

OV-5 describes the operational activities that are normally conducted in the course of achieving a mission or an objective. It describes capabilities, operational activities (or tasks), dependencies between activities, and dependencies to/from activities that are outside the scope of the architecture. An example OV-5 is not provided in this paper.

### **6.2.6 Operational Activity Sequence and Timing Description (OV-6)**

Many of the critical characteristics of architecture are only discovered when the dynamic behavior of the architecture elements is modeled to incorporate sequencing and timing aspects of the architecture. The dynamic behavior concerns

the timing and sequencing of events that capture the behavior of operational activities, thus describing an operational process or a mission. The same is true of SOA. OV-6 includes three products. They are:

- Operational Rules Model (OV-6a)
- Operational State Transition Description (OV-6b)
- Operational Event-Trace Description (OV-6c)

#### **6.2.6.1 Operational Rules Model (OV-6a)**

The OV-6a defines the operational policies and rules that govern operational resources. The policies and rules dictate what the resources must do and what they must not do and therefore affect how operational activities are conducted. An example OV-6a is not provided in this paper.

#### **6.2.6.2 Operational State Transition Description (OV-6b)**

The OV-6b is a graphical model that describes how an operational resource or activity responds to various events by changing its state. An example OV-6b is not provided in this paper.

#### **6.2.6.3 Operational Event Trace Description (OV-6c)**

OV-6c is an essential process (control) flow model used to describe a mission thread (a sequence of operational activities, or business process flow). In a SOA, the focus is on the operational process, i.e., the sequence of operational activities that provide the context and rationale for creating the services. OV-6c is essential for SOA as it describes the operational process requirements used in the allocation and design of human and system services in the SV. An example OV-6c is not provided in this paper.

#### **6.2.7 Logical Data Model (OV-7)**

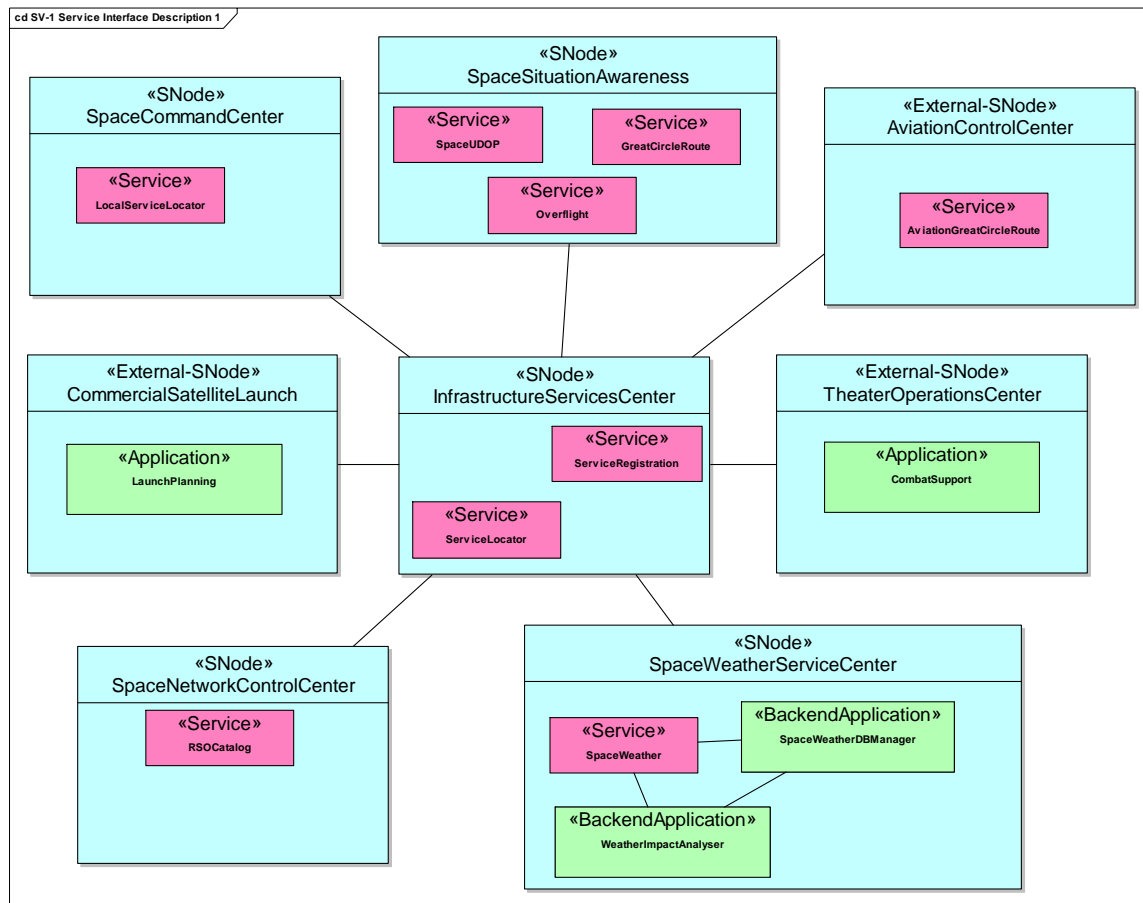
OV-7 describes the structure of an architecture domain's information and data types and the structural operational rules that govern the data. It provides the architecture domain vocabulary, taxonomy, and upper-level ontology for data consumed and produced by services. A common vocabulary that is agreed upon by the domain's CoI is key in resolving data semantic issues leading to information systems interoperability. An example OV-7 is not provided in this paper.

### **6.3 Systems View (SV)**

SV consists of a set of graphical and textual products that describe human functions and automated system processes (services) and their interconnections in support of DoD operational activities. The relationship between architecture data elements across the SVs to the OVs can be described as follows: humans and automated systems, or human functions and services, are grouped into nodes and fielded to provide capabilities described as OV requirements and to execute operational activities. Some tailoring of the SV is required to support SOA. Because of this paper's scope of **software** services, the example SV products do not include the human view.

#### **6.3.1 Systems Interface Description – (SV-1)**

SV-1 shows where services are desired by describing logical or physical nodes that group services and human functions. Services and human functions may be logically grouped by where they are advertised (e.g., all services published in a particular registry) or by some broad functional category (warfighting, logistics, financial management, HR, medical, etc.); SV-1 also shows any planned or actual logical communication channels (connectors). Figure 1 shows the major nodes and services for the sample scenario. Systems node connectors indicate that the infrastructure services center is where services are registered by providers and then located by consumers.



**Figure 1: SV-1 Diagram Showing Services Grouped by Systems Nodes and Their Dependency on the Infrastructure Services Node**

Figure 2 is an SV-1 diagram that shows service providers and service consumers. The example is for illustration purposes only and is not a complete SV-1 diagram. The services shown are the same ones that are defined in SV-4 diagrams. The connections across service specifications are intended to illustrate that after locating a service, a potential consumer connects directly to the service provider through the exposed service specification (*provided* service specifications are indicated by the circle notation, *required* service specifications are indicated by the cup notation). The connections are provided here for illustration purposes only. It is neither required nor useful to show every single possible connection across service providers and consumers.

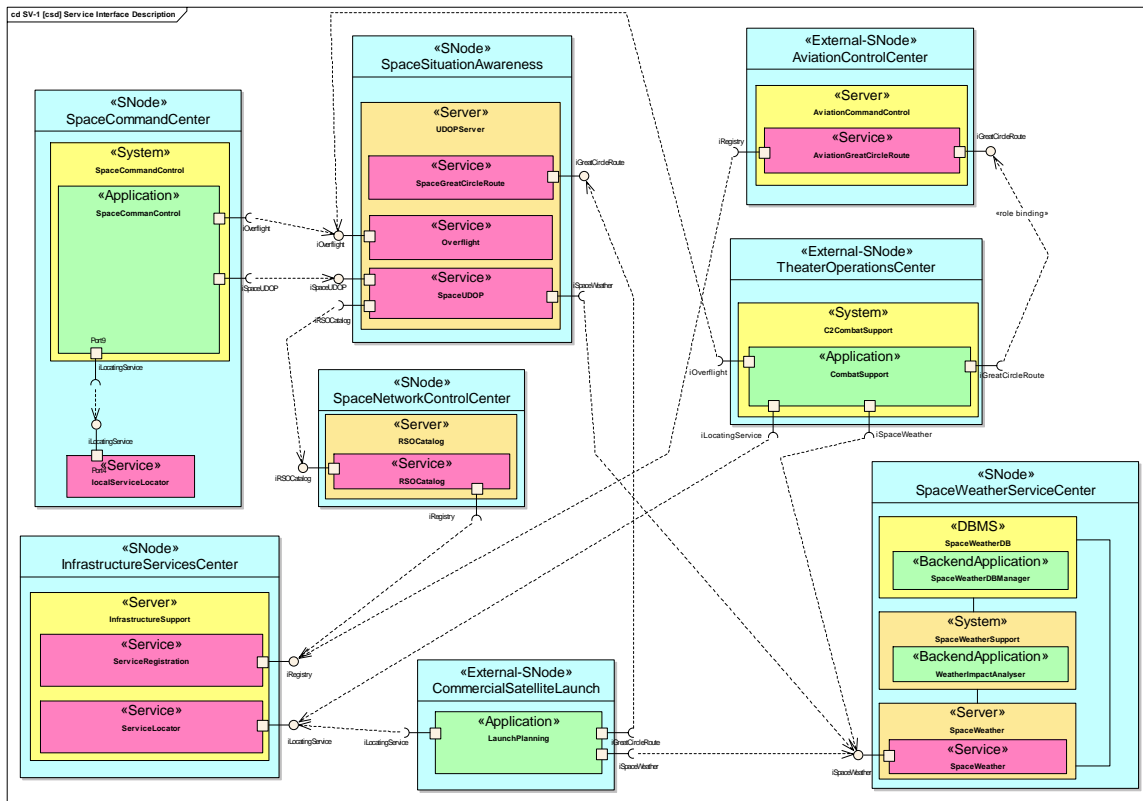


Figure 2: SV-1 Showing Systems Nodes, Systems, Services, and Service Interactions

### 6.3.2 Systems Communication Description – (SV-2)

SV-2 shows where the hosting centers are, and what service implementations they are hosting. It also shows where the major points of presence are on the network, and where the various groups of human users (surrogates for information requestors) are on the network. SV-2 can specify how each SV-1 logical systems node connector is physically implemented (e.g., the standards used). There is no example of SV-2 provided for this sample architecture.

### 6.3.3 Systems-Systems Matrix – (SV-3)

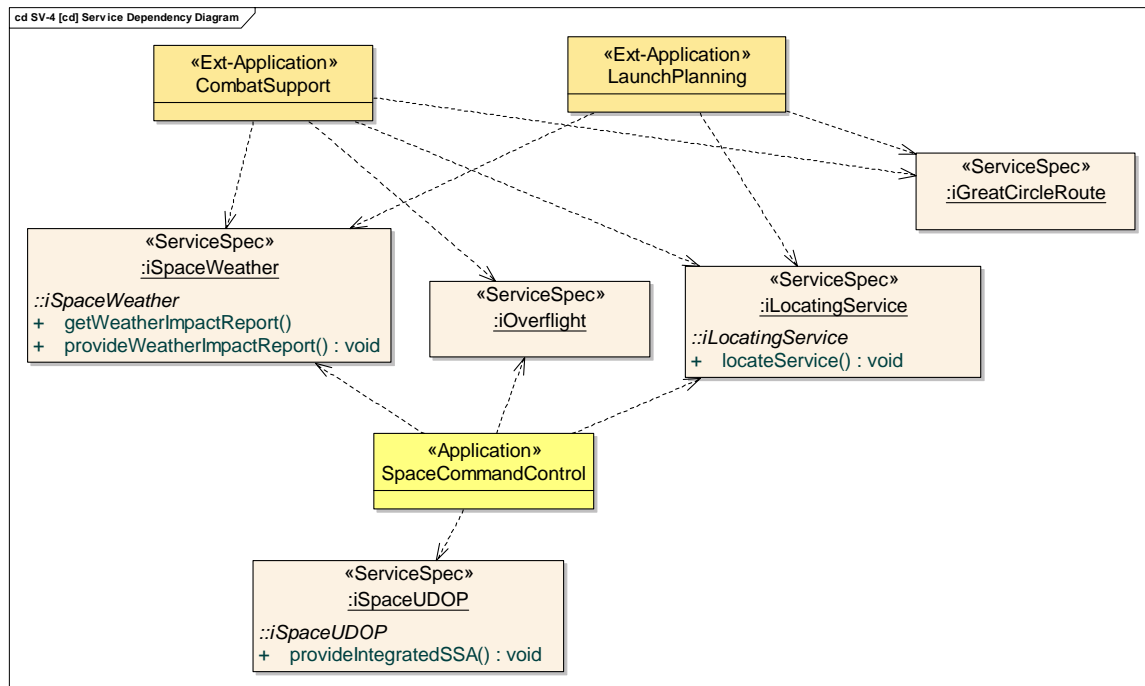
SV-3 provides detail on the systems interface characteristics described in SV-1 for the architecture. The product may be tailored to include a matrix that shows the dependency between users of services as rows, and the required services (through the service specification) as columns. Table 1 is an example of such a matrix showing the list of applications as users of services and the service specifications upon which they depend.

Table 1: Tailored SV-3 Showing Application to Service Dependencies

Application	iGreatCircle Route	iLocating Service	iOverflight	iSpace UDOP	iSpace Weather
CombatSupport	X	X	X		X
SpaceCommandControl		X	X	X	
LaunchPlanning	X	X			X

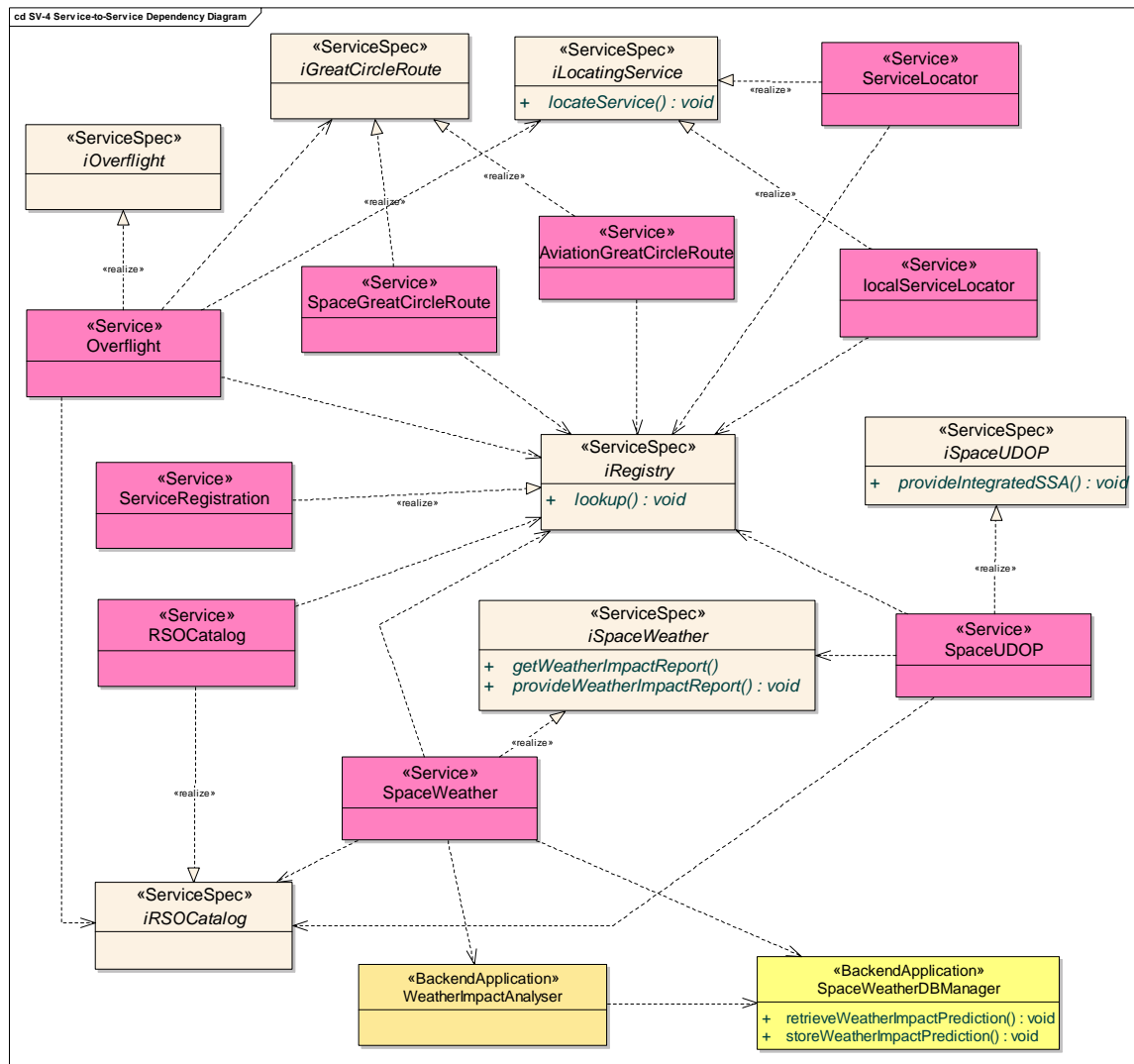
### 6.3.4 Systems Functionality Description – (SV-4)

SV-4 documents specifications of human functions, services, and their known product (data or material) flows, human and machine interactions. SV-4 shows flows between services and human functions and can be tailored to show service specifications, service realizations, and their dependencies as demonstrated in the following service dependency diagram. Figure 3 shows the various software service specifications and the applications that are dependent on those services through the exposed service specification.



**Figure 3: SV-4 Showing Services and Application Dependencies**

Another tailored SV-4 (see Figure 4) shows the service specifications, services (specification realizations), and their dependencies.



**Figure 4: SV-4 Showing Service specifications, Services, and Their Dependencies**

In addition to the above, a table that provides a *Service Specification Template* (SST) [12] entry for each service may be provided. The data that goes into a SST Entry is derived from the service attributes defined in SV-4, rules, policies, and effects defined in SV-10a & b, as well as quality performance requirements defined in SV-7. The SST provides a common reference model for defining and describing GIG enterprise service offerings to DoD Information Technology (IT) service providers, consumers, developers, and managers. The SST identifies information that either *must* or *can* be supplied to effectively describe a service on the GIG. This information includes, but is not limited to:

- What the service does
- How to access the service
- What security mechanisms or restrictions apply to the service
- Various points of contact for the service—including the service provider
- Service level characteristics
- Performance information for the service.

- Inputs, outputs, preconditions, and effects with respect to service operations [12]

### 6.3.5 Operational Activity to Systems Function Traceability Matrix – (SV-5)

SV-5 shows the traceability of requirements from OV's through their decomposition and allocation to SV's that satisfy the requirements. It provides a summary of those relationships in one easy-to-examine format. The SV-5 matrix may be generated from allocation relationships established between operational activities in OV-5 and human functions/services in SV-1 or SV-4 and can show the relationships between the set of operational capabilities and activities applicable to an architecture description as allocated to the set of systems and human functions/ services that apply to that architecture description. Table 2 is an SV-5 that shows the mapping of operational activities to corresponding human functions and/or services.

**Table 2: SV-5 Showing Operational Activities and Corresponding Human Functions and/or Service**

Operational Activity	Human Function (Role)	System Function/Service
Monitor Space Environment	Provide Space Weather Service	SpaceWeather
	Maintain Space Sensor	RSOCatalog
Monitor Space Force Operation	Provide Space UDOP Service	SpaceUDOP
	Supply UDOP criteria	
	Watch UDOP	
Provide Theater Operations Support	Provide Overflight Service	Overflight
	Provide Space Weather Service	SpaceWeather

### 6.3.6 Systems Data Exchange Matrix – (SV-6)

SV-6 is a Systems Data Exchange Matrix that documents data exchanges between two pairs of functions performed by systems and can be tailored as a Services Data Exchange Matrix generated from the services and their dependencies as described in SV-4 and SV-10c. A partial example SV-6 is provided below. However, this example does not include all attributes that can be used to describe the characteristics of a service data exchange.

**Table 3: Tailored SV-6 Showing Services Data Exchanges and Their Attributes**

Service Name	Service User	Data Required	Data Produced	Transaction Type
Space UDOP	SpaceCommandControl Application	UDOP criteria	UDOP	Publish/ Subscribe
SpaceWeather	LaunchPlanning Application	date-timeRange, location	spaceWeatherImpactReport	Request/Respond
SpaceWeather	CombatSupport Application		spaceWeatherReport	Broadcast

### 6.3.7 Systems Performance Parameters Matrix – (SV-7)

SV-7 specifies performance characteristics of systems and hardware/software items. SV-7 is important to SOA because it can be tailored to capture the performance requirements of services identified in a service level agreement, and allows

for a comparison between specified performance thresholds and actual performance levels achieved by service implementations during testing. A partial example SV-7 is provided below.

**Table 4: Tailored SV-7 Showing A List Of Services And The Quality Of Service Supported By The Provider**

Service	Quality Metric	SLA		Service Quality
		Threshold	Objective	
Space Great Circle Route	Availability	99%	100%	99.5%
Space Great Circle Route	Response Time	.001 second	.0001 second	.002 second
Aviation Great Circle Route	Availability	95%	100%	97%
Aviation Great Circle Route	Response Time	2 second	.1 second	1.2 second

### 6.3.8 Systems Evolution Description – (SV-8)

SV-8 captures evolution plans that describe how the systems (one can include the system functions and service implementations performed by the systems) will evolve over a period of time. SV-8 describes the changes required or planned in service specifications, and required or planned changes to systems. An example SV-8 is not provided in this paper.

### 6.3.9 Systems Technology Forecast – (SV-9)

SV-9 defines the underlying current and expected supporting technologies. New technologies should be tied to specific time periods, which can be correlated against the time periods used in SV-8 milestones. SV-9 describes the required or planned changes to technologies specified in service specifications, and the changes to technologies that support service implementations and deployment. Any predicted or proposed changes to technologies related to the SOA being architected will need to be identified so a plan to incorporate or otherwise address the changes can be put into place. An example SV-9 is not provided in this paper.

### 6.3.10 Systems Functionality Sequence and Timing Descriptions – (SV-10)

SV-10 describes system solutions to the requirements described in OV-6, and describes the dynamic behavior of humans and systems, and of human functions and services. Three types of models may be used to adequately describe the dynamic behavior and performance characteristics of a Systems View. These three models are:

- Systems Rules Model (SV-10a)
- Systems State Transition Description (SV-10b)
- Systems Event-Trace Description (SV-10c)

SV-10b and SV-10c may be used separately or together, as necessary, to describe critical timing and sequencing behavior in the SV.

#### 6.3.10.1 Systems Rules Model – (SV-10a)

SV-10a describes human function policies and procedures, and service policies and rules. These include policies and rules related to service level agreements, as well as behavioral rules associated with the orchestration of services. The SV-10a generally flows from the OV-6a. An example of SV-10a is not provided in this paper.

### 6.3.10.2 Systems State Transition Description – (SV-10b)

SV-10b is a graphical model that describes state changes on humans and systems performing human functions or services. The diagram basically represents the sets of events to which the systems in the architecture will respond (by taking an action to move to a new state) as a function of its current state. An example SV-10b is not provided in this paper.

### 6.3.10.3 Systems Event-Trace Description – (SV-10c)

SV-10c describes the sequence (control) flows for human functions and services. For SOA, these service sequence flows can be used to specify service orchestrations that will need to be implemented as rules in an orchestration engine, or they can be used to specify the sequence of flows between human functions, application services, and infrastructure services. Figure 5 is an activity diagram that details the process flow among interacting humans, services, and applications.

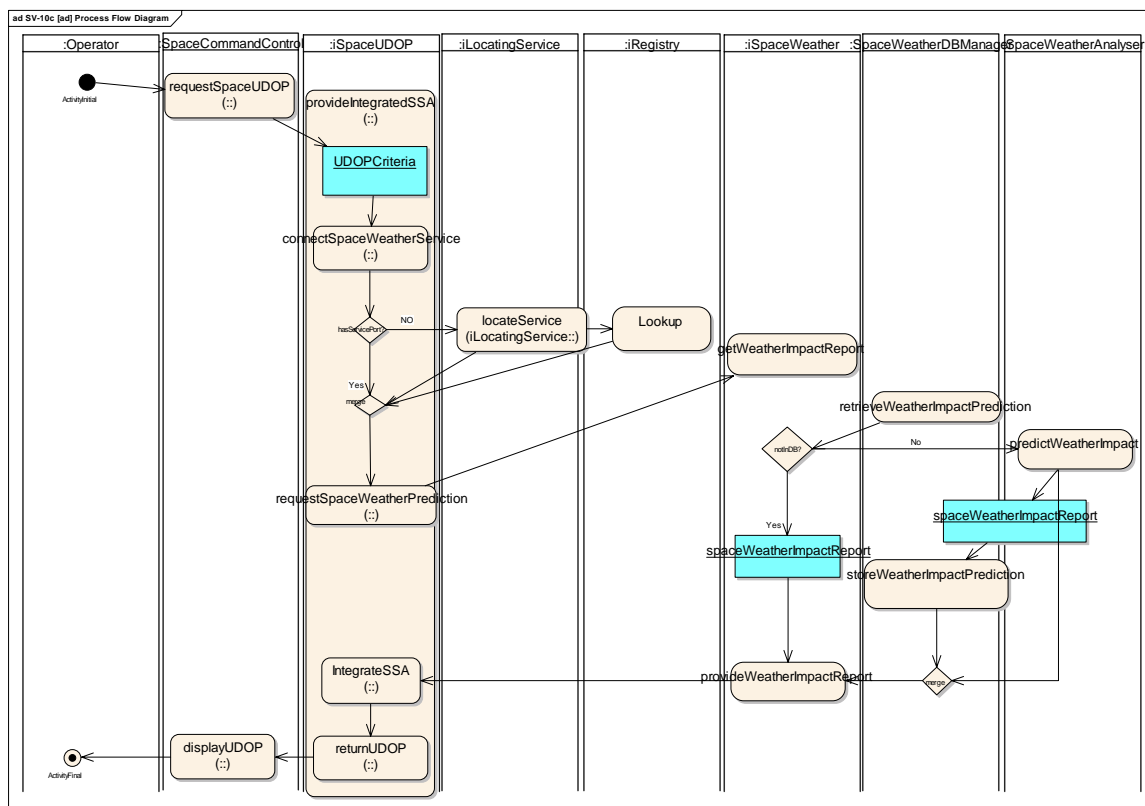
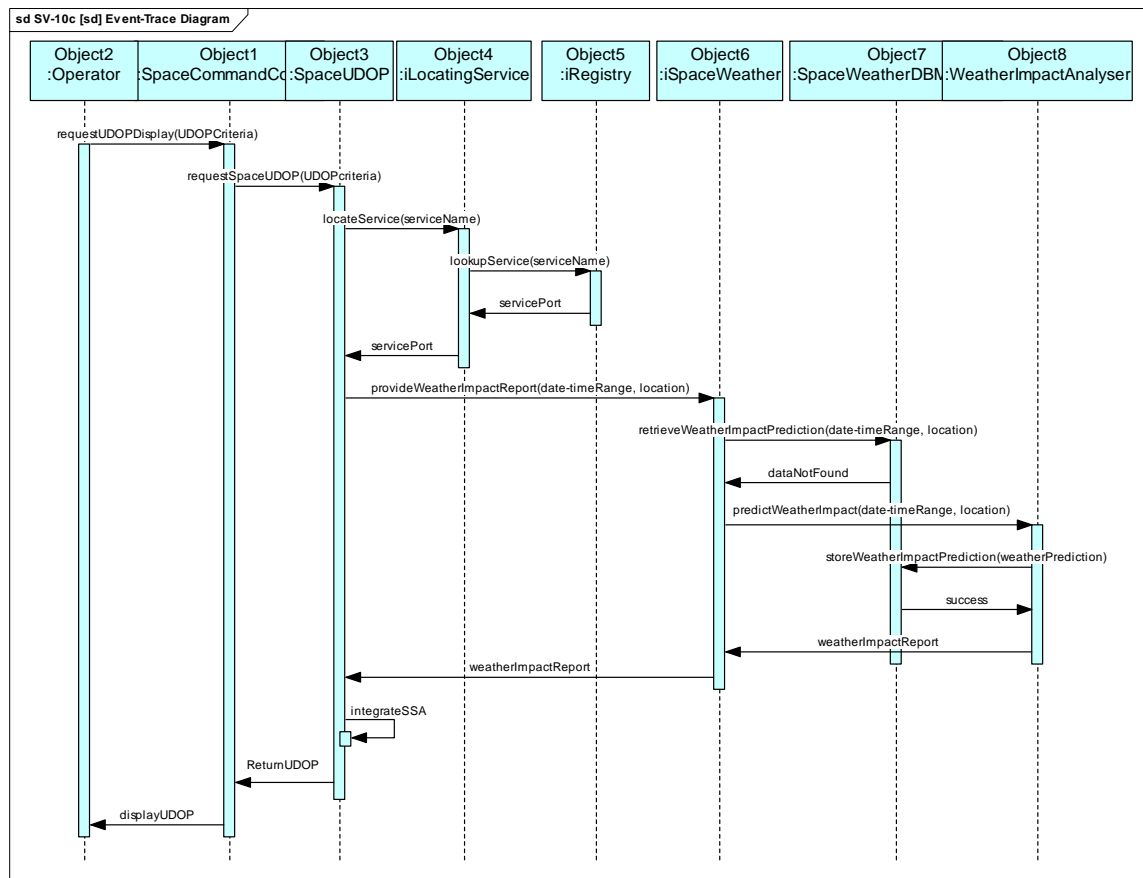


Figure 5: SV-10c Showing a Process Flow with an Activity Diagram

Figure 6 illustrates a particular scenario of the process flow illustrated in Figure 5.



**Figure 6: SV-10c Showing a Scenario with a Sequence Diagram**

### 6.3.11 Physical Schema – (SV-11)

SV-11 describes the structure of the various kinds of domain data that are utilized by the systems in the architecture. An example SV-11 is not provided in this paper.

## 6.4 Technical View (TV)

The TV provides the technical implementation standards upon which engineering specifications are based, common building blocks are established, and product lines are developed. Standards used in specifying and implementing services are described in TVs. There are two key TV products. The TV-1 defines the current defined baseline for the architecture, while the TV-2 defines upcoming and future standards.

### 6.4.1 Technical Standards Profile – (TV-1)

TV-1 lists the various standards rules that implement and sometimes constrain the choices that can be made in the design and implementation of an architecture. Primarily, this is a generated product that summarizes the standards rules and conventions that apply to architecture implementations. The profile may be time phased to facilitate a structured, disciplined process of system development and evolution. Time-phased migration promotes the consideration of emerging technologies and the likelihood of current technologies and standards becoming obsolete. An example TV-1 is not provided in this paper.

#### 6.4.2 Technical Standards Forecast – (TV-2)

TV-2 is a detailed description of emerging technology standards relevant to the enterprise service offerings covered by the scope of the architecture. It can contain predictions about the availability of emerging standards and the likely obsolescence of existing standards in specific time frames (e.g., 6-month, 12-month, 18-month intervals), and confidence factors for the predictions. It also can contain matching predictions for market acceptance of each standard and an overall assessment of the risk associated with using the standard. The forecast includes potential standards impacts on current architectures, and thus influences the development of transition and objective architectures. The forecast should be tailored to focus on technology areas that are related to the purpose for which a given architecture description is being built, and should identify issues that will affect the architecture. TV-2 summarizes the standards rules and conventions that apply to SV-9 elements according to SV-8 milestones. An example TV-2 is not provided in this paper.

## Appendix B

Below is a summary of the terms used in this paper, as well as excerpts from a DoDAF metamodel showing relevant SOA tailoring. The new elements are *service port* or interaction point, *service specification* or service description, and *service requirements* or contract. Definitions for these new elements are provided below. Table 5 provides a quick summary of the terms used and corresponding terms from DoDAF and industry.

**Port.** Specifies a distinct interaction point between an element of the architecture and its environment. It specifies the services provided as well as the services required by that element. Service specifications are provided and required through ports in order to decouple consumers and providers.

**Service Specification.** A specification of what service consumers need to do in order to use a service. A service provider defines the service specification.

**Service Requirement (Contract).** Agreed upon functionality description that a service must meet. It includes a description of the participants in the contract and the *roles* they play.

**Table 5: Summary Of Terms Used And Corresponding Terms From Dodaf And Industry**

Term used in Paper	OASIS Term	OMG (draft) Term	DoDAF Element (specialized)
Performance Requirements	N/A	Service Level Agreement (SLA)	Measure of Performance (MOP)
<b>Port</b>	Interaction	Interaction Point	N/A
Service	Service	Service	System Function
<b>Service Specification</b>	Service Description/Interface	Service Specification	N/A
<b>Service Requirement</b>	Contract	Contract	N/A
Standard	Standard	Standard	Technical Standard
Data	Real World Effect	Service Data	Systems Data

Figure 7 and Figure 8 show relationships between the new elements and between these elements and other existing DoDAF elements. Elements that are new to DoDAF v 1.0 are highlighted in Blue. Relationships to and from these elements constitute new relationships.

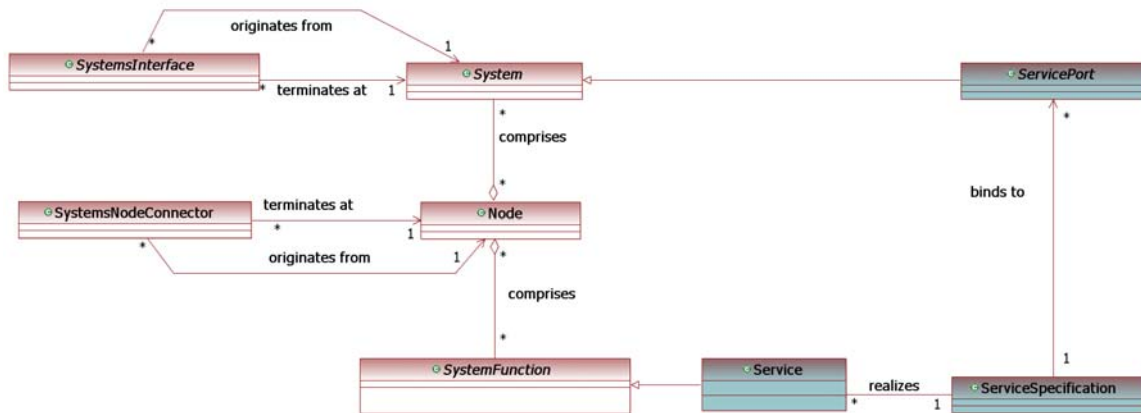


Figure 7: Metamodel Excerpt for SV Elements and Relationships

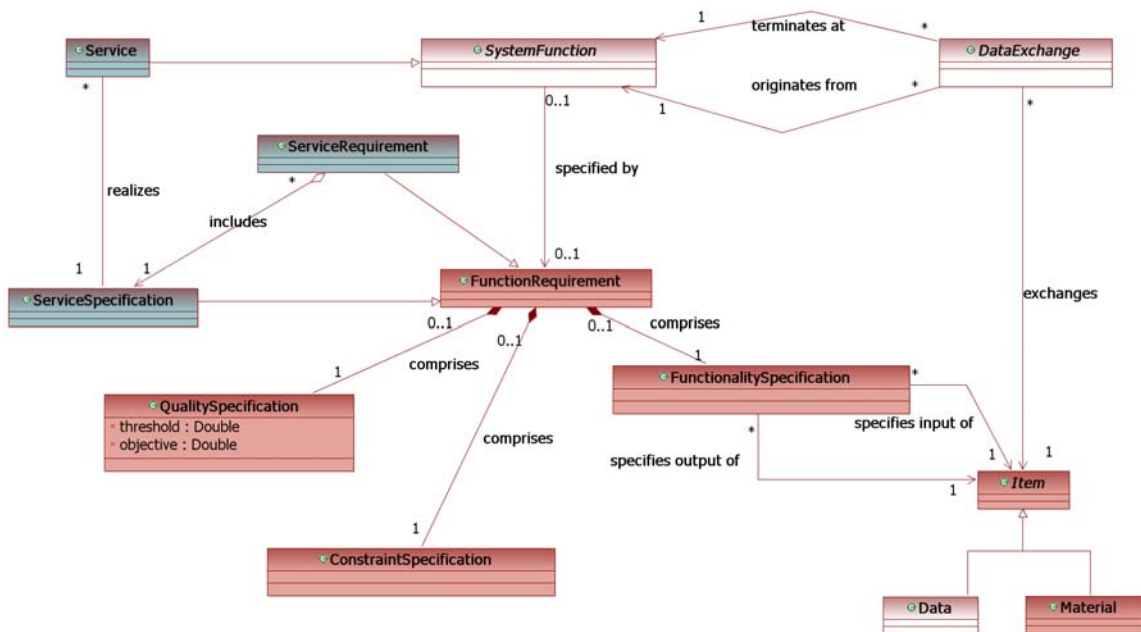


Figure 8: Metamodel Excerpt for SV Elements and Relationships